

In the claims

1. (Currently Amended) A system for managing transactions between ~~a first computer system and a second computer system~~ computer systems, comprising:

a receiving computer system that receives a request message from a first computer system, wherein the request message corresponds to a response having one or more transaction types and having one more attributes for each of the one or more identified transaction types;

a management file accessible by the receiving computer system and containing at least one a plurality of transaction type-attribute strings having a mask format with a corresponding action value with multiple transaction type-attribute strings of the plurality comprising a transaction type identified by a transaction code having one or more characters masked so as to represent multiple transaction types, the multiple strings further comprising an attribute and an associated action value representing an action to be taken by the receiving computer system for the attribute, wherein the management file contains a reduced number of transaction type attribute strings with respect to a fully populated management file not containing transaction type attribute strings with the transaction code having one or more characters masked;

a process implemented by the receiving computer system that determines what action to take in accordance with the at least one transaction type-attribute string in response to the received request message by comparing one of the corresponding transaction codes with the transaction type of the plurality of transaction type-attribute strings of the management file to find a transaction type-attribute string that matches both the transaction type and one of the attributes corresponding to the received request, and by obtaining the action value associated with the transaction type-attribute string that has been the found.

2. (Currently Amended) The system recited in claim 1, ~~further comprising~~ wherein the receiving computer system is a gateway computer to which the first and a second computer systems are connected, and through which the first and second computer systems communicate with one another, wherein the process is executed on the gateway computer.

3. (Original) The system recited in claim 2, wherein the manager file is loaded into a memory residing on the gateway computer at runtime.
4. (Previously Presented) The system recited in claim 1, wherein the management file further contains an additional at least one transaction type-attribute string having a corresponding action value, the additional at least one transaction type-attribute string not having a mask format.
5. (Currently Amended) The system recited in claim 4, wherein the process looks for a literal match to the transaction type prior to looking for a non-literal match, ~~the match being made with a match transaction type-attribute string determined by the process and wherein a literal match is one with no characters masked and a non-literal match is one with at least one character masked.~~
6. (Currently Amended) The system recited in claim 5, wherein the process performs an action determined by the transaction type-attribute string that is a non-literal match and that has ~~having the minimum mask matching the match transaction type-attribute string.~~
7. (Currently Amended) A method for managing transactions between a ~~first computer system and a second computer systems~~, comprising the steps of:
- (a) creating a management file containing ~~at least one~~ a plurality of transaction type-attribute strings ~~having a mask format and a corresponding action value with multiple transaction type-attribute strings of the plurality comprising a transaction type identified by a transaction code having one or more characters masked so as to represent multiple transaction types, the multiple strings further comprising an attribute and an associated action value representing an action to be taken by the receiving computer system for the attribute~~, wherein the management file contains a reduced number of transaction type attribute strings with respect to a ~~fully populated~~ management file not containing transaction type attribute strings with the transaction type having one or more characters masked;

- (b) receiving an inbound transaction request;
- (c) processing the inbound transaction request to determine a transaction type and an attribute for the transaction type that corresponds to the inbound transaction request;
- d) based on the determined transaction type, determining a response transaction type-attribute string having no masked characters and having an appropriate attribute for responding to the inbound transaction request;
- (e) determining if the response transaction type-attribute string having no masked characters matches one of the at least one transaction type-attribute strings in the manager file; and
- (f) if a match is found, responding to the input transaction request in accordance with the action value associated with ~~the action value corresponding to the~~ transaction type-attribute string that matched the response transaction type-attribute string.

8. (Previously presented) The method recited in claim 7, further comprising the steps of:

- (g) populating the management file with at least one additional transaction type-attribute string with a corresponding action value, the at least one additional transaction type-attribute string not having a mask format;
- (h) determining if the response transaction type-attribute string literally matches one of the at least one additional transaction type-attribute strings; and
- (i) if there is a literal match, responding to the inbound transaction request in accordance with the action value associated with one additional transaction type-attributes strings that matched.

9. (Currently Amended) The method recited in claim 8, further comprising the step determining if there is literal match prior to determining if there is a match requiring a mask, where a literal match is one with no masked characters.

10. (Original) The method recited in claim 8, further comprising the step of generating an error message if there is no transaction type-attribute string that matches the response transaction type-attribute string.

11. (Previously Presented) The method recited in claim 7, further comprising the step of storing the management file in a computer memory at runtime.

12. (Currently amended) The method recited in claim 8, further comprising the step of using the transaction type-attribute string having the minimum mask but still matching the ~~match-response~~ transaction type-attribute string.

13. (Original) The method recited in claim 8, further comprising the step of repeating steps (a) through (i), for each attribute in a response type corresponding to the input transaction request.

14. (Currently Amended) A system for managing computer transactions between a first computer system and a second computer system, comprising:

a gateway computer providing communication between the first computer system and the second computer system[[]];

a process, which when executing on the gateway computer, takes a transaction request from the first computer system, reformats the transaction request into a request the second computer system can process, transmits the reformatted transaction request to the second computer system, receives a response transaction from the second computer system, reformats the response transaction into a format the first computer can process, at least a portion of the response transaction being based on a table ~~having one or more transaction type-attribute strings having a mask format, wherein the table contains a reduced number of transaction type-attribute strings with respect to a fully populated table~~ containing a plurality of transaction type-attribute strings with multiple transaction type-attribute strings of the plurality comprising a transaction type identified by a transaction code having one or more characters masked so as to represent multiple transaction types, the multiple strings further comprising an attribute and an associated action value representing an action to be taken by the receiving computer system for the attribute, wherein the table contains a reduced number of transaction type attribute strings with respect to a table not containing transaction type attribute strings with the transaction type having one or more characters masked.

15. (Original) The system recited in claim 14, wherein the table is stored in a memory on the gateway computer.

16. (Original) The system recited in claim 15, wherein the table is stored in memory at runtime.

17. (Original) The system recited in claim 14, wherein the table includes at least one additional transaction type-attribute string that does not have a mask format.

18. (Original) The system recited in claim 17, wherein the process determines a response transaction type-attribute string from the response transaction type string and one of the transaction type strings in the table.

19. (Currently Amended) The system recited in claim 18, wherein the process looks for a literal match prior to looking for a non-literal match, wherein the literal match is one with no characters masked.

20. (Currently Amended) A data structure stored in a computer memory of a receiving computer, comprising: a plurality of transaction code-attribute pairs, each transaction code-attribute pair comprising a transaction code portion and an attribute portion and being associated with an action value, wherein at least one of the plurality of transaction code-attribute pairs contains a mask value within the transaction code so as to represent a plurality of transaction code-attribute pairs and thereby reduce the number of transaction code-attribute pairs that are stored in computer memory, and wherein one transaction code-attribute pair has a most literal correspondence to a particular request sent from a first computer such that an action value for the particular request is determined by finding the transaction code-attribute pair having the most literal correspondence.

21. (Original) The data structure recited in claim 20, further loaded into the computer memory at run time.

22. (Previously Presented) The data structure recited in claim 20 further comprising a plurality of transaction code attribute pairs having a mask value in the transaction code portion, wherein the mask value of at least one of the transaction code portions of the plurality of transaction code attribute pairs having a mask value in the transaction code portion replaces only a portion of the transaction code portion.